AD-A088 595 NAVAL RESEARCH LAB WASHINGTON DC F/G 17/7
SOME PERFORMANCE RESULTS FOR RECURSIVE MULTITARGET CORRELATOR-T--ETC(U)
JUL 80 W W WILLMAN
UNCLASSIFIED NRL-8423 NL

1 OF 1
AD A
088595

END
DATE
FILMED
10-80
DTIC

LEVEL II (12) 8

(14) NRL-~~Report~~-8423

(6) **Some Performance Results for Recursive Multitarget Correlator-Tracker Algorithms.**

(10) WARREN W. WILLMAN

Systems Research Branch
Space Systems Division

(11) 23 Jul 80

(12) 24

(9) Final rept.,

July 23, 1980

(16) F12143

(17) XF12143310

251950

NAVAL RESEARCH LABORATORY
Washington, D.C.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Report 8423	2. GOVT ACCESSION NO. AD-A088	3. RECIPIENT'S CATALOG NUMBER 595
4. TITLE (and Subtitle) SOME PERFORMANCE RESULTS FOR RECURSIVE MULTITARGET CORRELATOR-TRACKER ALGORITHMS		5. TYPE OF REPORT & PERIOD COVERED Final report on one phase of a continuing NRL problem
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Warren W. Willman		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, DC 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62712N; XF12-143-310; 79-0724-0-0
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Electronic Systems Command ELEX-03 Washington, DC 20360		12. REPORT DATE July 23, 1980
		13. NUMBER OF PAGES 23
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Ship tracking	Adaptive driving noise
Ocean surveillance	Contact correlation
Multitarget tracking	Recursive filtering
Kalman filtering	Command and control

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A certain relatively simple type of recursive multitarget correlation and tracking algorithm is developed for the case of position-only target reports. Several aspects of its performance are tested numerically on simulated data resembling ship tracking situations of various target densities. Some simplifications of this procedure are also tested in this way to assess the relative contributions of several implementation concepts in this context.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

SOME PERFORMANCE RESULTS FOR RECURSIVE MULTITARGET CORRELATOR-TRACKER ALGORITHMS

INTRODUCTION

One common source of error and uncertainty in tracking several or an a priori unknown number of targets arises when this tracking is based on discrete reports, each giving information about the location of a target at a specified time but not about its identity. Uncertainty then exists as to which partition of the reports received is the one corresponding to the equivalence relation of target identity. Many schemes have been developed for dealing with this uncertainty, which is called the correlation problem, and the potential for complication is great [1]. This report examines the performance of a particular choice of algorithm when applied to some simulated report data resembling a ship tracking situation. The algorithm chosen is relatively simple but of a type that might eventually prove useful in actual applications for reasons given in the next section.

The nomenclature adopted here is that a target is an actual object in a surveillance region, whereas a track is an estimate of some target's past history produced by a correlator-tracker algorithm from the available report data. In the restricted context of this investigation, in fact, a track is always equivalent to some subset of these reports. The process of grouping reports together into such tracks is called correlation, association, or linking. A scan is a set of reports, all referring to the same time, for which set it is known that no two reports are of a single target.

Any reasonable correlator-tracker algorithm will operate properly in a sparse situation, and any algorithm will break down in a sufficiently crowded one. Thus the meaningful questions in this regard are how crowded a situation an algorithm can handle, how it breaks down when it does, and how well it indicates the resulting uncertainties when they do exist. The property of crowdedness here is composed of several factors, including high target density, infrequent reports, highly maneuvering targets, and high target velocities, all of which contribute to the difficulty of the correlation problem. These questions are investigated here only to the limited extent of testing the variations in several aspects of performance with target-density alone.

The reports that constituted these test data each specified a time, position, and position covariance matrix for an unidentified target but no other information. Hence correlation occurred on the basis of proximity to projected track positions alone. In real surveillance situations, at least some identifying target-attribute information is typically available also. Such a situation was not considered here for simplicity, but also it was hoped that considering the case of totally unidentified reports would be meaningful anyway, because the effect of attribute information would often be to decompose the problem into a number of subproblems, each with mutually indistinguishable targets but with a lower effective target density.

Manuscript submitted April 24, 1980.

Because the results obtained here are limited to a particular type of scenario and correlator-tracker algorithm, both described more fully in later sections, the nature of these results is more indicative than definitive. However, they do at least provide one specific case for possible future comparisons. Also, several variants of this algorithm were tested in this same context to provide clues about the importance of certain implementation concepts, such as using target-velocity estimates, using within-scan association constraints, and accounting for the effects of past correlation uncertainty.

RATIONALE FOR THE TYPE OF RECURSIVE PROCEDURE SELECTED

Consideration was limited here to simple algorithms, because these are relatively easy to implement and investigate in a flexible manner. However, it was also felt that some intrinsic interest might arise for nonoptimal but simple correlator-trackers because of their more modest requirements for computational resources. This would make them more widely dispersible for possible lower-capability local backup to a more sophisticated central facility.

The particular type of simple algorithms selected for investigation here was singled out because it operates naturally in real time on a time-ordered stream of input reports and because it can be constructed as an extension of the recursive multitarget tracker described in Fig. 1 for the case of independent target motion and no correlation uncertainty (the target identity is known for each report). This simple case reduces to an independent single-target tracking situation for each target present. Hence the correlation step (step 3) of the algorithm in Fig. 1 is trivial, and steps 2 and 4 are performed independently for each track (target in this situation) by a single-target tracker. The tracker used for this purpose is the Kalman filter with adaptive driving noise described in Refs. 2 and 3, which has been reported to give fairly consistent results in sea trials [4]. The target-status parameters updated by this filter are the first 20 listed in Table 1 and constitute the information that must be carried along for each track in order that the algorithm can operate in the recursive, real-time manner shown.

The algorithms considered here for the case of reports *without* specified target identity all operate as shown in Fig. 2 and differ only in the approximations used in some of the computations mentioned there. This operation is otherwise of the same form as that of Fig. 1, except that the correlation step (step 3) is no longer trivial and is split into the three parts shown. Also, in one of the variants considered here, the list of target-status parameters is augmented slightly to include the position, error-ellipse parameters, and probability of correct association for the most recent report in the track. The function of all the variants considered here of this basic procedure can be summarized as follows:

Input: A time-ordered sequence of instantaneous *scans* of an area of interest. Each scan is a set of *reports* such that no two come from the same target. Each report gives the 2σ ellipse for the position of a target at the *time* of the scan, which is also an input.

1. Read the reports in the current scan.

2. Project the target-status parameters of existing tracks to the current time.*

3. Append each report of a target formerly seen to that (unique) existing track which consists of reports from that target. (Each remaining report is from a distinct target not formerly seen, since the correlations are known.)

4. Update the target-status parameters of each track augmented by a current report in step 3.*

5. Enter each remaining current report as the first report in a new track and set the initial target-status parameters for that track.

continue to next scan

*Steps 2 and 4 are performed independently for each track by an adaptive Kalman filter.

Fig. 1 — Operation of the recursive multitarget tracking algorithm for independent target motion and *no correlation uncertainty*. Operation begins with no currently existing tracks.

Table 1 — List of Target-Status Parameters

1.	Time of the latest report in the track.
2-15.	Mean and covariance matrix (four by four) of the conditional distribution, given all the available data, of the *current* position and velocity of the target generating the *latest* report in the track.
16,17.	Estimates of the in-track and crosstrack intensity components of the driving noise describing the maneuvers of this target.
18.	Estimated time of the first report of this target in the track.
19.	Estimated number of reports of this target in the track which contain significant information about the driving-noise intensity.
20.	Limiting parameter used in updating the driving-noise estimates.
21.	Probability that the last two reports in the track came from the same target, given the available data.
22-26.	Mean and covariance matrix (two by two) of the target position in the latest report in the track.

1. Read the reports in the current scan.

2. Project the target-status parameters of existing tracks to the current time.

3a. Compute* the association likelihood for each pairing consisting of a current report and an existing track.

3b. Select the most likely* pairing (and reports to be left unpaired) for which at most one report is appended to each existing track.

3c. For each such linked report, compute the probability that the most recent report in the assigned track came from the same target.

4. Update* the target-status parameters of each track augmented by a current report.

5. Start a new track with each unlinked current report and set the initial target-status parameters.

continue to next scan

*Based on certain approximations necessary to retain this recursive form.

Fig. 2 — Operation of the recursive multitarget tracking algorithm for independent target motion and correlation uncertainty

Output (in real time):

- A set of *tracks*, each a time-ordered sequence of past input reports. The set of tracks always forms a *partition* over these reports.
- The current values of the set of *target-status parameters* for each track.
- The probability (conditioned on currently available data) that for each pair of successive reports in each track both reports came from the same target.

User-specified parameters:

- The 90th percentile of the target speeds.
- The area of the surveillance region.
- The elapsed time since the most recent report to declare a track lost (after which no new reports can be added to it).

- The minimum time between *significant* target maneuvers (minimum elapsed time over which the dead-reckoned position, using actual instantaneous initial position and velocity, would have an error outside the average report error ellipse in 10% of the cases).

The target identities of the reports are not specified here, so the assignment of new reports to old tracks (or as the first reports of new tracks) are necessarily based on proximity to dead-reckoned track positions alone. As the density of targets increases, therefore, there is increasing uncertainty as to which partition of the reports is the one corresponding to the equivalence relation of target identity (where the set of false alarms, if any, is regarded as a target). This so-called correlation uncertainty has the following consequences.

- The correspondence between targets and the tracks generated by the algorithms here is no longer certain or, in general, perfect. Hence it now makes a difference whether the target-status parameters are defined in terms of the first or most recent (or other) report in a track.

- The probability distributions of the target-motion state (current position and velocity), conditioned on past reports but marginal over correlation hypotheses, are no longer independent for different tracks, even when the motions of distinct targets are independent. The target corresponding to a given track here is of course defined as that generating the last report assigned to that track.

- Even the marginal distribution of the target-motion state of a single track, given the data, is of a complicated multimodal form because of the multiple correlation hypotheses possible. This is true even if these distributions are Gaussian whenever conditioned on a particular correlation hypothesis. These consequences of correlation uncertainty make the accurate computations of steps 3 and 4 of Fig. 2 tremendously complicated even under the ideal circumstances of complete knowledge of the statistical parameters governing the tracking situation, independent target motion, and single-target motion and sensor information which corresponds to a standard Kalman filtering format. (References 5 and 6 describe these ramifications in more detail.)

All the variants of the algorithm considered here perform approximations to these computations which depend on only the parameters of the current scan of reports and the current values of the target-status parameters (listed in Table 1) for each current track. This feature makes them suitable for real-time operation. Also, the pairing procedure of step 3b in Fig. 2 is always done by the matrix-scan method. This method can be explained by imagining the results of step 3a being arranged in a matrix of likelihoods where rows correspond to current reports and columns correspond to existing tracks, with an extra column for the likelihood of each report being from a new target. The pairing is then generated by first selecting the report-and-track pair corresponding to the largest likelihood entry in the entire matrix, then deleting that row and column (or just the row if the new track option is selected) and repeating the process with the reduced matrix until all the reports or all the existing tracks are exhausted. All remaining reports are deemed new tracks; all remaining tracks are simply not processed further on that scan. This matrix-scan procedure was selected because of its computational simplicity and because it has been shown experimentally in Ref. 7 to give (usually) a good approximation to that pairing which

maximizes the product of the likelihoods of the pairs chosen when the number of reports and tracks are equal and exactly one report must be paired with each track.

The other approximations of steps 3 and 4 are based on various simplifying assumptions which are described in more detail in the next section. In each variant of the algorithm, these simplifications are such that step 3 produces the *a posteriori* most probable assignment of new reports to existing tracks and new targets for independent target motion if:

- The matrix-scan procedure succeeds in maximizing the product of likelihoods,

- There was no correlation uncertainty in any of the preceding scans (so that the motion-state distributions for the existing tracks are independent, and hence so that maximizing the product of individual-pair likelihoods maximizes the joint likelihood of the entire pairing),

- The prior probability distribution of the number of new targets detected in the current scan is uniform on $\{0, 1, ..., n\}$, where n is the number of reports in the scan, and

- The conditional density of the position of each existing track, given the data of preceding scans, is Gaussian with parameters as given by the operation of steps 2 and 4 acting as a single-target tracker.

Because of this property at the time of the *first* scan with correlation uncertainty, it is hoped that the effects of such uncertainty will die out quickly, at least when it is small, thereby causing these algorithms to operate nearly optimally.

To summarize, the particular kind of recursive correlator-tracker algorithm selected for examination here was chosen because it is simple, operates naturally in real time, and gives the most likely association of new reports with existing (and new) tracks for the first scan at which correlation uncertainty is present. Also, the algorithms seem to provide a type of output which would be useful in a variety of situations. Figure 3 summarizes the operation of this kind of correlator-tracker schematically.

THEORY AND ALGORITHMS

Since the algorithms here are based on a number of simplifying approximations, the theory behind them was used only as a rough guide to their construction, which was refined by a certain amount of numerical experimentation with various theoretically plausible options. Care was taken, however, to refrain from using strictly ad hoc adjustments that were found to improve performance, in order to avoid the phenomenon of tuning the algorithms to the particular data at hand.

All of the algorithms tested here reduce to the same single-target tracking procedure in the absence of any correlation uncertainty. This procedure is the same as that specified in Ref. 3, except for two minor modifications to improve numerical efficiency and stability. First, the estimates of the maneuvering-noise intensity and the velocity covariance submatrix

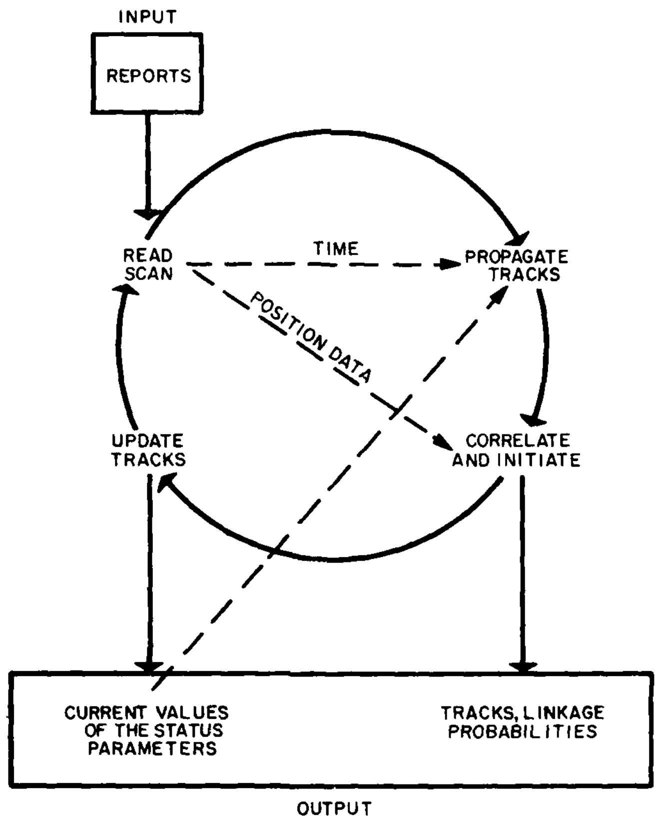

Fig. 3 — Summary of the recursive-tracker operation

are not updated by a new report in the track (and the report is not counted as containing significant information for such updates in the future) if the time since the last update is shorter than a user-specified minimum. This parameter is the elapsed time over which the dead-reckoned position of a randomly selected target at a randomly selected time, using its actual initial position and velocity, would have an error (due to target maneuvers) outside the 2σ ellipse of a randomly chosen report with a probability of 0.1. Second, some initial maneuvering noise is provided by the second report in a track containing significant maneuvering information by setting the intensity of both noise components, and augmenting the velocity covariance submatrix, as if about 30% of the observed position change arose from random maneuvering.

Since all the algorithms are based on this same single-target tracker, they all initialize the target-status parameters of new tracks in the same way (step 5 of Fig. 2). For those parameters pertaining to single-target tracking (parameters 1 through 20 in Table 1) the initial values are those described in Ref. 3 for single-target tracking, where parameters 18, 19, and 20 are initially the current time, 1, and 0 respectively. The other initial parameter

values are obvious, except for the probability of correct association, which is superfluous initially and is arbitrarily set to zero.

Conceptually tracks are never deleted here. Once created, they always remain candidates for further augmentation. They just never are after their target locations are projected outside the surveillance region or become so diffusely distributed that other associations are always chosen for new reports. As a practical matter of numerical efficiency, however, a track is automatically barred from further augmentation if the elapsed time since its last update with a new report ever exceeds a user-specified threshold.

The way in which the linkage probabilities are computed in step 3c, given the likelihoods from step 3a, is also the same in all the algorithm variants examined here. This method is based almost entirely on the case of a scan with a single report. In this case the number n of new targets present in the surveillance area since the last scan (more precisely, the number of targets which will eventually be detected at least once that have not yet been) is estimated as

$$n = \frac{sm}{T},$$

where

m = number of tracks already created,
s = elapsed time since the last scan, and
T = elapsed time since tracking began.

The theory here is that target arrivals are approximately a Poisson process and that most of the old targets that will ever be seen have been seen. Then, taking the current target population as definitely n new targets plus a distinct target for the latest report in each existing track, and regarding the (single) report in the current scan as being emanated by each of these putative targets with equal probability (before conditioning on the report's data values), the posterior probability of the report-to-track linkage of step 3b being correct is approximated as

$$P_c = \frac{g_i}{\frac{n}{A} + \sum_j g_j}, \tag{1}$$

where

A = area of the surveillance region,
i = index of the track paired with the report, and
g_j = likelihood (from step 3a, in the same area units) of the report data, given that the report came from existing track j

and where the index j is over all previously existing tracks. This is just the probability that would be given by the Bayes rule if these simplifying assumptions were correct and if newly

detected targets were equally likely to be reported anywhere in the surveillance region (so that the likelihood of the reported position is 1/A).

For the case of multiple-report scans, this same procedure is applied to each report independently, *except* that the quantities g_j being summed in Eq. (1) are all replaced by $\max(g_i, g_j)$ to provide at least some crude compensation for the fact that the within-scan constraints on report-to-track linking are being disregarded. The reasoning here is that if the likelihood g_i (for the track i with which the report in question has been paired) is less than g_j for some other report j, then the reason that this other pairing was not made is that the within-scan linking constraints would require that track j be unlinked from some other report in the scan, leading to a set of feasible linkings of lower overall posterior probability. Hence, for any report considered in isolation, its posterior probability of actually coming from (the target giving the last report of) the track to which it has been linked by the matrix-scan procedure, which does take the constraints into account, should be an upper bound on that of its coming from any other track. If these posterior probabilities were computed according to the right-hand side of Eq. (1) for *each* track i, then this modification would at least retain this property if the scan constraints caused the linkage of the report in question to some track other than the one giving the maximum-likelihood value for that report. It was considered too complicated at the outset to consider any procedure that examines all significantly probable permutations of new reports among existing and new tracks or examines joint likelihoods of such pairings. Some other more sophisticated possibilities for approximating P_c in multiple-report scans were tried that still did not require these complications, but none was found which worked any better than the one just described.

These values of P_c are also used for the linkage probabilities in the output, although these should really also be conditioned on the reports in subsequent scans. It was felt that so many approximations had already been made that such a refinement would be superfluous. This judgment appeared to be borne out by trials of a limited updating method of this sort which updated values of P_c with report data from only the next scan at which the target-status parameters of the track in question were updated.

It now remains only to describe how steps 2, 3a, and 4 of Fig. 2 are performed. For a given scan in each case, step 2 proceeds by projecting the first 20 target-status parameters (of Table 1) for each existing track with the single-target tracker as described at the beginning of this section, using the updated or initialized parameter values from the preceding scan. The other six track-status parameters are left unchanged at this point. After this the various options examined here are all some form of simplification of the following procedure for steps 3a and 4.

For each pair consisting of a current report and an existing track being considered for association, two likelihoods g_A and g_B are computed. The first is computed from the projected target-status parameters of step 2 for that track as if the assignment from the preceding scan were indeed correct, the conditional target state density were Gaussian with the corresponding mean and covariance matrix, and the report errors were Gaussian. Hence

$$g_A = \frac{e^{-(z - \overline{x})^T (M + R)^{-1} (z - \overline{x})}}{2\pi |M + R|^{1/2}}, \tag{2}$$

where

z = reported position (two-vector),
R = covariance matrix (two by two) of a Gaussian density with the same 2σ ellipse as that reported,
$\bar{x}$ = mean of the projected track position (two-vector), and
M = covariance matrix (two by two) of the projected track position.

$\left.\begin{array}{l}\bar{x}\\ M\end{array}\right\}$ subset of the target-status parameters

The other likelihood is computed as if the last report assigned to the track in question came from a new target, so that g_B is given by Eq. (2), with $\bar{x}$ being the position given by the last report in this track (two other target-status parameters) and with

$$M = R_0 + \left[\begin{array}{c|c} \frac{V}{2}(t - t_0)^2 & 0 \\ \hline 0 & \frac{V}{2}(t - t_0)^2 \end{array}\right], \tag{3}$$

where

R_0 = covariance matrix associated with this last report,
t_0 = time of this last report,
t = time of the current scan, and
V = square of the 90th percentile of the target speeds.

$\left.\begin{array}{l}R_0\\ t_0\end{array}\right\}$ target-status parameters

This second likelihood is now used not only for the hypothesis that the last report in the track is from a new target but also for all possible associations other than the one actually selected, because the specific track identities of these other past alternatives are not retained. Hence g_B really corresponds more closely to a target-motion hypothesis than an association hypothesis. According to this approximation, then, the overall association likelihood for the report-and-track pair in question here becomes

$$g = p_c g_A + (1 - p_c) g_B, \tag{4}$$

where p_c is the probability of correct association computed by Eq. (1) (or its modification) for this track in step 3c of the scan in which it was last updated. The probability p_c is another target-status parameter unless the track is new (contains only one report so far), in which case $g_A = g_B$ and the value of p_c does not matter. The resulting value of g from Eq. (4) is the value for this pair used in steps 3b and 3c of the current scan.

After completion of the other parts of step 3, the updating step (step 4) proceeds by first merging the first 20 target-status parameters for the two target-motion hypotheses of each track selected in step 3b and then updating them with the single-target tracker as described at the beginning of this section. Except for the covariance matrix parameters, this merging is just a weighted average of the two values of each parameter for the two

target-motion hypotheses considered, the weight for the hypothesis corresponding to correct linkage at the *last* update being p_c, as in Eq. (4) for the likelihoods. The merged position-velocity covariance matrix M is given by

$$M = p_c[M_A + (\bar{x}_A - \bar{x})(\bar{x}_A - \bar{x})^T] + (1 - p_c)[M_B + (\bar{x}_B - \bar{x})(\bar{x}_B - \bar{x})^T], \tag{5}$$

where $\bar{x}$ is the merged mean (four-vector), the subscript A denotes the parameter values for the correct previous linkage, and the subscript B denotes the parameter values for the incorrect previous linkage. Since the single-target tracker here operates basically as a Kalman filter, this merging procedure effectively amounts to approximating the two-term Gaussian sum density corresponding to the above two target-motion hypotheses by a standard (four-variate) Gaussian density with the same mean and covariance matrix. The updating of the last six parameters in Table 1 is self-explanatory from their descriptions there, except that p_c is now replaced by the value computed from Eq. (1) (or its modification for multiple-report scans) in step 3c of the *current* scan.

The following simplifications of this overall procedure were also tested here, the idea being to learn something about the significance of some of its features for performance:

A. The alternate target-motion hypothesis is not considered in performing the correlation step (step 3). This simplification is achieved here by merging the target-status parameters (as described above, but for all tracks now) as the end of step 2 instead of the beginning of step 4. The likelihoods in step 3a are then computed as the right-hand side of Eq. (2) with $\bar{x}$ and M now denoting the merged parameters for the track in question. The rest of the updating in step 4 proceeds as before.

B. The merging procedure at the beginning of step 4 is not performed, meaning that the alternate target-motion hypothesis is not used in updating the target-status parameters.

C. This alternate motion hypothesis is not used in either the correlation step or the updating step. All three of the preceding variants reduce to this same simplification if p_c is made unity for each track treated in steps 2, 3a, and 4.

D. Target-velocity estimates are not used. This simplification is implemented by always using $p_c = 0$ in step 3a, which means that target motion is always treated as that of a new track (which has no established velocity). In this case, only the first and the last five target-status parameters listed in Table 1 are important, and the projection and updating of steps 2 and 4 (Fig. 2) are largely trivial. This simplification is sometimes advocated, especially for rarely detected submarine targets [1]. In this context, however, it seems intuitively that forming target-velocity estimates would be helpful, and this is indeed the case.

Roughly speaking, simplification A here corresponds to neglecting the effects of previous correlation uncertainty in current correlations, simplification B corresponds to neglecting these effects in track updating, and simplification C corresponds to ignoring correlation uncertainty entirely (except in computing the linkage probabilities of step 3c for output). The unsimplified algorithm includes these effects, to a limited extent, in both the tracking and correlation operations.

DESCRIPTION OF THE TEST DATA

The algorithms were tested numerically on simulated report data from a surveillance region 200 by 300 n.mi. in extent. The overall areal density of ships was varied, but for each density about 90% of them were simulated as being in lanes whose total area was about 7000 n.mi.^{2}. The ship speeds were fairly evenly distributed over a range of 5 to 25 knots in each case, and about 10% of the ships were simulated as moving erratically in the manner of fishing vessels or combatants.

The generated reports each contained a time, an error-corrupted position of an unidentified ship in the surveillance region at that time, and the parameters of the 90% confidence ellipse for this error. Conditions here were such, however, that these position errors were nearly always a minor factor affecting performance. Also about 5% of the reports were of ships detected only once, so these essentially acted as false alarms in forming nontrivial tracks (tracks with two or more reports). Two types of reports were generated on the basis of the simulated shipping:

- Reports from instantaneous scans of the surveillance region approximately every 2 hours such that no two reports in the same scan came from the same target. The set of reports comprising each such scan was assumed known to the tracker. Except for ships detected only once, the detection probability for each ship was greater than 90% in such scans.

- Sporadic, independent reports for each target, with an average time between reports of about 5 hours.

The numerical tests were made with such input data simulated over an interval of 14 hours, so most ships present in the surveillance area at the end were not there at the beginning, and vice versa. Also, for each shipping density simulated, each algorithm variant was tested over the same set of input data. About 3/4 of the reports in each case were of the first type described (scan data with high detection probability). Reports of the second type were treated by the algorithms here as if they were scans consisting of one report each.

PERFORMANCE STATISTICS

Several summary statistics were computed from the outcomes of these tests for each algorithm variant and each shipping density simulated. These statistics were chosen to give a meaningful indication of how well the algorithms performed as correlator-trackers. However, most of these performance indicators are specialized for the particular type of algorithm studied here and in many cases to the specific type of input data as well. The simulations were terminated immediately after one of the scans of reports was generated (of the first type of data). These statistics were then applied only to targets reported at least once in the time interval between and including the last two such scans and to tracks updated at least once during the same interval. Since these last two scans happened to be only 1-1/2 hours apart, and since the target detection probability is high for both of them, this gives a good approximation to the set of targets actually present when the simulation ends. Likewise, this selective subset of tracks is something a reasonable observer might use as an estimate of the past history of this set of targets.

One concept used in generating these statistics is *track depth*. This is defined as the number of reports one can go back from the most recent report, in a track created by an algorithm, before the actual target identity of a report changes, or before the first report of the track is reached. Thus this depth is an integer from zero to the integer that is one less than the number of reports in the track. A track's depth is zero if it contains only one report, or if the last two reports in it actually come from different targets due to imperfect performance of the tracker-correlator. An *estimated track depth* is also computed here from the algorithms' outputs. This is the number of reports one can go back in a track before the product of the corresponding linkage probabilities p_c, as generated by the algorithm in question for the adjacent reports in that track, becomes less than 1/2. Comparisons of the actual and estimated track depths give an indication of how well the linkage probabilities generated by an algorithm correspond to its actual performance. For the subsets of targets and tracks examined here, the data were such that the tracks formed by a perfect tracker would have an average depth of about 4 (five reports per track), and about 85% of the tracks would have two or more reports (depth $\geqslant 1$). The most meaningful distinction of track quality for the range of imperfect performance observed here was taken to be whether its depth was zero or nonzero. The reason for this is that the last two position reports (which are always fairly accurate here) can be used to form a fairly accurate target-velocity estimate in the latter case but not the former, and this in turn is important for localizing the track at future times.

For only the restricted subsets of targets and tracks mentioned earlier the following quantities involving track depth were computed for each algorithm tested and shipping density simulated:

- The fraction of those tracks formed by the algorithm which had nonzero actual depth.

- The weighted-average estimated depth of those tracks (the sum of all the depths divided by the number of tracks). This statistic can be computed without a knowledge of ground truth. Thus it would be useful for real applications if this were a reliable indicator of actual performance.

- *Only for those tracks with nonzero estimated depth*, the weighted-average actual track depth. The reason for this extra restriction is to make the measure more sensitive and to provide some penalty for optimistic estimates of track depth.

- The preceding average as a fraction of the weighted-average *estimated* depth over the same set of tracks. The ideal value is unity. Lower values indicate overly pessimistic estimates of track depth, and higher values indicate overoptimism.

- The determinant of the "classification matrix" for the two categories of zero and nonzero track depths. This can be computed as

$$\frac{ad - bc}{(a + b)(c + d)},$$

where

a = number of tracks of nonzero actual depth which are estimated to be such,
b = number of tracks of nonzero actual depth whose estimated depth is zero,
c = number of tracks of zero actual depth whose estimated depth is 1 or greater, and
d = number of tracks of zero actual depth whose estimated depth is also zero.

This determinant provides a single summary of an algorithm's classification performance for these two categories of track quality. Its value is always between +1 and −1. It attains the value +1 for perfect classification (as long as there is at least one track actually in each category), −1 for completely wrong classification, and 0 if track depths are estimated as zero or nonzero at random with equal probability or if all tracks are estimated to be in the same category.

One possible use for correlator-tracker output might be to direct intensive surveillance (such as aircraft) to all currently present targets that have been detected. With the assumption that such surveillance can be applied quickly, the performance of this function mainly depends on how well the set of current targets corresponds to the set of currently active tracks generated by the correlator-tracker. To test this ability, the following two performance statistics were also computed, again only for the restricted set of targets and tracks described earlier:

- The percentage of these targets for which no report is assigned by the algorithm as the latest report in one of these tracks. This can happen if reports from this target get mixed in the same track with those from another.

- The percentage of these targets for which reports are assigned as the last report of more than one track (which always turned out to be two here). This can happen if a report fails to be linked (say by erroneously starting a new track) with the preceding report from the same target.

The idea here is that intensive surveillance might reasonably be directed at the set of dead-reckoned positions corresponding to the latest report in each track of this restricted set. Hence these two percentages would correspond to missed targets and duplicated efforts. Unless the presence of the within-scan association constraints is ignored, they themselves keep the percentage of missed targets from being too large here. Most of the targets ever reported in this restricted time interval are reported in the two scans at either end of it, and each of these scans contains a report from each of about 3/4 of these targets. Hence the percentage of missed targets cannot rise much above 25%.

Table 2 lists the values of these seven performance indicators that were obtained experimentally with the test data described in the last section for a variety of algorithm variants and shipping densities. Also, they are listed for the unsimplified variant operating on the same input data but without making use of the within-scan correlation constraints. These same results are displayed graphically in Fig. 4, except that the missed-target and duplicated-target percentages are added to give a single combined figure of merit for this kind of performance, under the assumption that the two kinds of errors are of roughly equal importance.

INTERPRETATIONS AND COMMENTS

These results pertain directly only to the performance of the correlation function: to the association of reports into tracks that correspond to targets, not to the localization of these targets (the tracking function). In this context, however, correlation decisions are based on proximity to projected track positions alone, so that the performance of each of these functions depends on the performance of the other. Also, a more detailed examination of some of the output showed that its overall quality corresponded pretty much as might be expected to that indicated by these summary statistics. For example, when bad performance was indicated, track segments would be generated crosswise to shipping lanes almost as often as along them, whereas most linkages would be in the direction of a lane when these indicators were good.

The statistical reliability of these results was not analyzed, but the degree of raggedness of the curves of Fig. 4 gives a rough indication of it. There it no reason to think that these curves would not be smooth and monotonic except for sampling error. Hence only the gross features of the curves are likely to be meaningful. Because of this and also the restricted context and range of algorithm options studied here, conclusions drawn from the results given in Table 2 and Fig. 4 should be regarded as indicative rather than definitive.

With this caveat, plausible smooth curves can be fitted to the data of Table 2 and Fig. 4 to give the following general conclusions:

- Making use of within-scan association constraints and forming target-velocity estimates are important. Target-velocity estimates are mainly helpful at the lower target densities, but high densities are less important because performance there is poor anyway.

- It is helpful to include the effects of previous correlation uncertainties in current correlation decisions and track updating. To the limited extent that these effects were included here, they typically resulted in a level of performance that could be reached at only about half the target density required for that level when past correlation uncertainties were ignored. In cases such as this with fairly accurate reports, it seems reasonable to postulate that a dimensionless performance indicator would depend primarily on average target speed V, and average time T between reports of a target, and density D of targets. Dimensional analysis would then show that such an indicator would be a function only of the product DV^2T^2. Hence the inclusion of previous correlation uncertainty here would alternatively allow the reporting frequency to be reduced by a factor of about $\sqrt{2}$ while maintaining the same level of performance.

- Significant performance improvement results from taking account of past correlation uncertainty in both current correlation decisions and track updating. Its role is perhaps somewhat more important in track updating at high target densities (when performance is poor anyway) and in current correlation decisions at low densities.

- Even for the basic unsimplified algorithm, performance did not become generally good here until the overall target density D fell below about 2×10^{-4} n.mi.$^{-2}$. Since the average target speed V was about 15 knots and the average time T between reports of a target was about 1.8 h, the corresponding value of the difficulty parameter DV^2T^2 is about

Table 2 — Performance Statistics for Various Target Densities Up To a Maximum Target Density of 0.0015 n.mi.$^{-2}$

Performance Statistic	Algorithm Variant*	Maximum Target Density Divided by:						
		21	13	8	5	3	2	1
Fraction of the tracks with nonzero actual depth	1	0.81	0.82	0.73	0.75	0.62	.057	0.39
	2	0.83	0.83	0.65	0.66	0.51	0.42	0.29
	3	0.83	0.80	0.74	0.64	0.58	0.55	0.37
	4	0.79	0.78	0.78	0.54	0.55	0.39	0.34
	5	0.73	0.84	0.59	0.55	0.45	0.39	0.25
	6	0.82	0.78	0.69	0.69	0.61	0.58	0.36
Determinant of the classification matrix (for zero vs nonzero depth)	1	0.82	0.75	0.58	0.44	0.42	0.58	0.41
	2	0.69	0.67	0.48	0.37	0.46	0.50	0.47
	3	0.86	0.78	0.60	0.38	0.26	0.40	0.36
	4	0.77	0.80	0.60	0.31	0.35	0.31	0.27
	5	0.62	0.57	0.49	0.41	0.37	0.46	0.33
	6	0.71	0.60	0.46	0.37	0.28	0.66	0.25
Weighted-average actual depth for tracks with nonzero estimated depth	1	4.1	4.0	3.6	2.9	2.8	3.0	1.7
	2	3.8	3.8	3.1	3.0	3.1	2.7	2.4
	3	4.1	3.7	3.9	2.9	2.4	2.2	1.6
	4	3.9	3.7	3.6	2.3	2.4	1.7	1.1
	5	3.3	3.2	3.0	2.6	2.3	2.6	1.3
	6	4.0	3.5	2.8	2.8	2.4	2.2	1.0
Fraction of the targets missed (not the latest report in any track)	1	0.02	0.02	0.07	0.07	0.08	0.15	0.17
	2	0.01	0.02	0.08	0.08	0.14	0.16	0.19
	3	0.02	0.02	0.05	0.06	0.07	0.14	0.22
	4	0.02	0.02	0.05	0.09	0.07	0.15	0.17
	5	0.04	0.02	0.10	0.14	0.15	0.17	0.20
	6	0.05	0.05	0.13	0.17	0.22	0.31	0.41
Fraction of the targets duplicated (latest report in two or more tracks)	1	0.02	0.02	0.06	0.04	0.03	0.05	0.03
	2	0	0.07	0.04	0.05	0.08	0.06	0.05
	3	0.02	0.05	0.03	0.05	0.07	0.03	0.07
	4	0.04	0.05	0.04	0.13	0.08	0.20	0.14
	5	0.01	0	0.05	0.07	0.09	0.06	0.05
	6	0.01	0.07	0.05	0.05	0.03	0.04	0.01
For tracks with nonzero estimated depth, the weighted-average actual depth as a fraction of the weighted-average estimated depth	1	1.01	0.97	1.07	1.04	1.01	1.22	0.83
	2	1.05	1.08	0.95	1.09	1.16	1.04	1.09
	3	1.05	0.96	1.06	1.06	0.99	0.95	1.00
	4	1.03	0.96	0.98	0.85	0.95	0.77	0.55
	5	0.96	1.04	1.12	1.08	1.08	1.50	0.88
	6	0.94	0.83	0.69	0.79	0.67	0.60	0.34
Weighted-average estimated depth over all the tracks	1	3.3	3.6	2.6	2.0	1.7	1.1	0.71
	2	3.0	2.9	2.3	1.7	1.2	0.78	0.49
	3	3.3	3.2	2.8	1.8	1.6	1.1	0.50
	4	3.1	3.2	2.9	2.2	1.8	1.3	0.96
	5	2.5	2.4	1.3	1.1	0.73	0.39	0.19
	6	3.6	3.7	3.5	3.0	3.0	3.1	2.3

*Algorithm Variants.
1 — basic algorithm,
2 — past correlation uncertainty ignored in current correlations,
3 — past correlation uncertainty ignored in track updating,
4 — past correlation uncertainty ignored in both,
5 — target velocities not estimated,
6 — within-scan correlation constraints ignored.

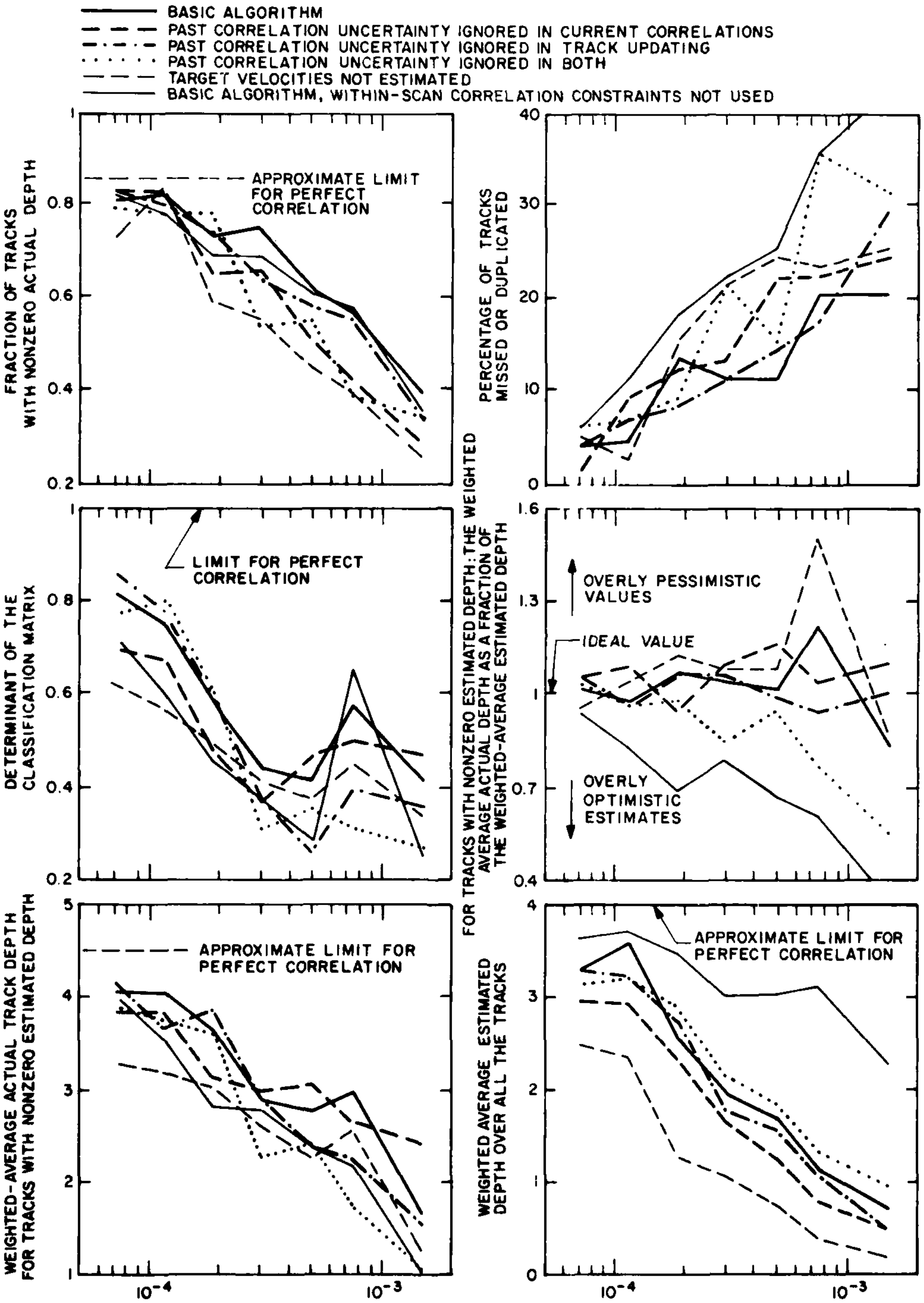

Fig. 4 — Performance statistics as a function of target density

0.15. This performance was mostly determined by conditions within the shipping lanes, however, where the great majority of the targets were confined. The corresponding value of DV^2T^2 for this restricted region is about 1, meaning a target density of about 1.3×10^{-3} n.mi.$^{-2}$.

- There is no evidence of a catastrophic cascading of errors when past correlation uncertainty is ignored entirely. This may be due to the adaptive-driving-noise feature used in the single-target tracking phase of the algorithms here. Incorrect reports associated with a track tend to make its corresponding target have a higher estimated maneuvering intensity, which compensates for the lack of explicit inclusion of the association uncertainty.

In the presence of within-scan association constraints, the level of performance could have been estimated reliably here, without knowledge of ground truth, by the estimated-average-track-depth statistic, which agreed fairly well with the actual average depth (not shown here for *all* tracks in the average). However, this statistic was only weakly dependent on actual performance in the absence of these constraints and became drastically over-optimistic at the higher target densities (lowest curve in the plot at the right center in Fig. 4). A possible explanation for this is that at higher densities the average spacing between targets began to be of the same order as the average error in reported positions, thereby reducing the basis for distinguishing between seeing fewer targets more frequently and more targets less frequently. An extreme example of this phenomenon would be if the reported position errors were so large that they provided no localization within the surveillance region at all. Then one would have only a sporadic series of reports that "a" target is detected "somewhere," and no information at all would be available to make this distinction in the absence of any additional specification of how often a target is likely to be detected or, alternatively, how many are likely to be present. For simplicity, no provision was made here for the former specification, because it was felt that such information would rarely be available. Scan-type data of high detection probability would provide the latter specification.

This explanation provides a possible remedy, because such a situation could be recognized *if* an algorithm's performance remains good enough long to be able to discern, without resort to ground truth, when the target density becomes high enough that the target spacing is less than the (known) report inaccuracy significantly often. In the scenarios examined here, this effect (as judged *with* a knowledge of ground truth) seems to become pronounced when the density rises above about 5×10^{-4} n.mi.$^{-2}$ in the absence of within-scan association constraints, as evidenced for example by the behavior of the curves for this case in the plot at the right center in Fig. 4. Figure 5, however, shows that estimates of target density are still roughly accurate at this point. Hence, even in the absence of scan-type data with high detection probability, this estimated density could be combined with a knowledge of report accuracies to reliably identify, without a knowledge of ground truth, cases in which the performance of this sort of algorithm is poor.

In the class of relatively simple algorithms here, essentially only the *level* of uncertainty in previous correlation decisions was ever taken into account in current correlation decisions and track updating, and even this was limited to the estimated uncertainty at the preceding updating step. A natural extension of this process is to retain the specific identities of these previous alternative associations and use alternate target-motion hypotheses which

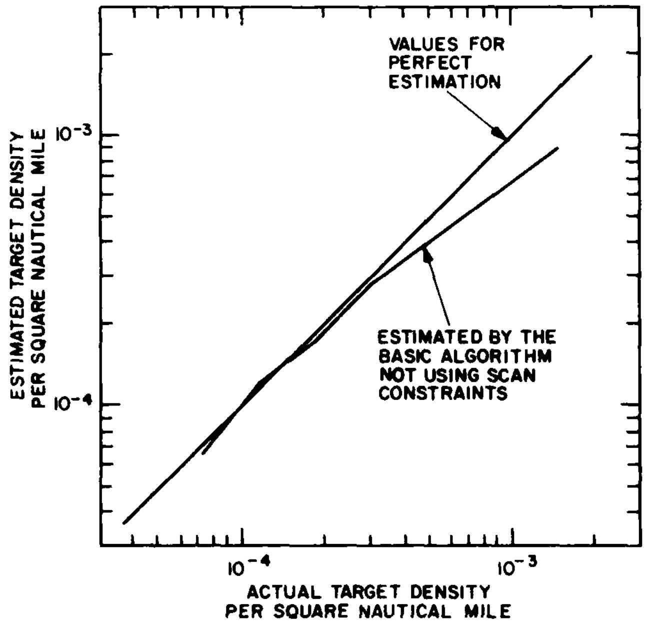

Fig. 5 — Accuracy of estimates of the target density

correspond more closely to these other options instead of blanketing them all with new target motion. This sort of extension would become very involved, however, especially when within-scan association constraints and possible new targets are taken into account, and was therefore beyond the scope of this investigation. If this sort of extension were done, though, there might be little to be gained for the kind of scenario considered here by extending the process further back in time. Tracking in a roughly similar situation, with mostly steadily moving targets and accurate position-only reports, was shown in Ref. 8 to improve substantially when such specific association alternatives from the first preceding updating step were included, with very little further improvement from including those from earlier steps as well. This is only to be expected in this context, however, because two successive reports determine a target's future positions fairly accurately, whereas one report alone fails to indicate velocity at all. Hence, if the degree of correlation uncertainty corresponds roughly to the fraction of overlap of confidence regions of projected target positions at the time of the next scan, this would lead to a great improvement, but the distinction between more remote alternatives would be only a small refinement. If this is indeed the governing mechanism here, however, one might obtain steady improvement up to three steps back with bearing-only reports, because four bearings are needed to pin down position and velocity and thereby future positions. On the other hand, if the target motions were highly erratic (but with low target density, so tracking were possible), then there would be little advantage to retaining specific associations even one step back with fairly infrequent position-only reports, because no useful target-velocity estimates could be formed anyway.

Actually two unsuccessful attempts were made to include this kind of processing to a limited extent. In the first attempt the alternative target-motion hypotheses projected the

target-status parameters as if the track in question had not been updated at all. The theory here was that this might be a closer approximation to the parameters for the target that *should* have been associated with that track. However, this procedure was found to work somewhat less well than the basic algorithm. One possible explanation is that using this other approach amounts to redefining the target-status parameters in terms of the target generating the last report that should have been associated with the track in question, which, since this is done at each updating step, is the same as defining the target-status parameters terms of the target generating the *first* report in the track. Thus the poorer performance may have resulted from an overemphasis on ancient history rather than emphasis on current data. The other attempt computed values of target-status parameters at the end of each scan for each track being updated which were merged values for each association possibility for the corresponding report. This merging, which was done *before* the next scan, proceeded in the same sort of way as in simplification A (described at the end of the section on theory and algorithms) but with more terms being merged. Therefore this merging resembled the procedure of Ref. 9 except that within-scan association constraints were ignored. Ignoring these constraints and doing this much merging apparently was too poor an approximation for this type of data, however, and resulted in catastrophic performance through the creation of ill-conditioned covariance matrices.

The importance of the startup problem and the effects of new targets entering the surveillance region was investigated by using the available target identities of reports in the simulated data here to insure that the first two reports of a new target were always the first two reports of a new track and vice versa. This produced a significant improvement in subsequent performance only when the target density was so high that performance was poor in either case.

The velocities of targets in a shipping lane are usually aligned with it. An attempt to exploit this information was made by altering the prior-velocity covariance matrix of each new track in a shipping lane so that the minor axis of the corresponding σ ellipse was perpendicular to the lane direction and 1/5 its original length. The resulting improvement in performance was only barely discernible, even though most of the reports in these examples came from targets in shipping lanes.

Except for input-output processing, which can vary widely with the application, FORTRAN implementations of the algorithms considered here typically used about 400 lines of code and processed about 15 reports per second on a PDP-11/70 computer.

ACKNOWLEDGMENTS

The author is grateful for assistance and helpful suggestions from L.J. Filippelli, H.L. Wiener, and I.R. Goodman.

REFERENCES

1. H.L. Wiener, W.W. Willman, J.H. Kullback, and I.R. Goodman, "Naval Ocean-Surveillance Handbook, 1978," NRL Report 8340, Oct. 31, 1979.
2. W.W. Willman, "Recursive Filtering Algorithms for Ship Tracking," NRL Report 7969, Apr. 6, 1976.
3. W.W. Willman, "Some Refinements for a Ship Tracking Algorithm," NRL Memorandum Report 3991, May 2, 1979.
4. T.G. Bugenhagen and L.B. Carpenter, "Over-the-Horizon/Detection, Classification and Targeting Engineering Analysis, Vol. 5, A Kalman Filter Rhumb-Line Ship-Tracking Algorithm," JHU/APL Report FS-79-032, Mar. 1979.
5. I.R. Goodman, "A General Model for the Multiple Target Correlation and Tracking Problem," pp. 383-388 in *Proceedings, 18th IEEE Conference on Decision and Control,* Dec. 1979.
6. I.R. Goodman, "A General Model for the Contact Correlation Problem," NRL Report 8417, 1980.
7. M.R. Fisher, "Study Concerning the Assignment Problem in a Ship Tracking Context," NRL Memorandum Report 2130, June 1, 1970.
8. R.A. Singer, R.G. Sea, and K.B. Housewright, "Derivation and Evaluation of Improved Tracking Filters for Use in Dense Multitarget Environments," IEEE Trans. on Info. Theory IT-20 (No. 4), 423-432 (July 1974).
9. Y. Bar-Shalom, "Extension of the Probabilistic Data Association Filter to Multitarget Tracking," pp. 16-21 in *Proceedings, 5th Symposium on Nonlinear Estimation Theory*, San Diego, July 1974.

DAT
ILM